Comments on the paper by J. Dörre, D. Gabbay and E. König: Fibred Semantics for Feature-Based Grammar Logic

Hans Leiß
CIS, Universität München

The aim of the paper is to investigate possible combinations of a "grammar logic", i.e., a reasoning method that establishes properties of strings – given properties of their atomic parts –, with a "feature logic", i.e. a system of constraints that are intended to somehow refine the grammar logic.

The paper begins with lengthy descriptions of Lambek's syntactic calculus (the example "grammar logic") and both a feature constraint logic and a feature term logic. It would have been sufficient to mention the completeness theorem for Lambek's calculus and the existence of a satisfiability test for the feature constraint logic. Also, the "singleton sorts" of Section 3 play a role only in the coding of 3SAT in Theorem 8, and the section about future work is about Lambek calculus without features.

The original contribution of the paper is contained in Sections 4 and 5, discussing semantics and proof theory of possible combinations of Lambek's logic with feature logic. Common to both combinations is the idea of fibred structures.

1 Fibred Structures

Let $\mathcal{S} \subseteq (2^{S^+}, \cdot, \rightarrow, \leftarrow, b^{\mathcal{S}})_{b \in BaseCat}$ be a standard (string) model for Lambek's calculus, with $\cdot$ as string concatenation on the set level, its residuals $\rightarrow$ and $\leftarrow$, and finitely many basic categories $b^{\mathcal{S}} \subseteq S^+$ as sets of nonempty strings over S. To fibre $\mathcal{S}$ basically means to split the syntactic categories of Lambek's calculus into various pieces, motivated by the classification by means of feature-value bundles used in unification based grammars. That is, $\mathcal{S}$ is to be expanded to a *fibred structure* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{F})$, where

i) $\mathcal{A}$ is a first-order (resp. feature) structure used to interprete terms t as elements (resp. subsets) $t^{\mathcal{A},g}$ of the universe $D^{\mathcal{A}}$ of $\mathcal{A}$, with respect to environments $g : Var \rightarrow D^{\mathcal{A}}$,

ii) $\mathcal{F} = \langle F_b \rangle_{b \in BaseCat}$ is a family of "fibring" relations $F_b \subseteq b^{\mathcal{S}} \times D^{\mathcal{A}}$, such that

$$b^{\mathcal{S}} = \bigcup \{ b(d) \mid d \in D^{\mathcal{A}} \}, \quad \text{where } b(d) := \{ \tau \in b^{\mathcal{S}} \mid \tau \, F_b \, d \}. \tag{1}$$

The structure on $D^{\mathcal{A}}$ is used to collect several $b(d)$'s according to properties of and relations between such $d \in D^{\mathcal{A}}$. Actually, the fibring relations F_b are not essential, but only that the objects d of the structure $\mathcal{A}$ define properties $b(d)$ of strings, related to $b^{\mathcal{S}}$ as expressed in (1). Hence, everything could also be done using *relations* $b \subseteq S^+ \times D^{\mathcal{A}}$ and then $b(d) := \{ \tau \in S^+ \mid \tau \, b \, d \}$ to define b^{S} if needed.[1]

[1] If one does not agree with the authors' argument against *replacing* atomic categories of Lambek's calculus by feature terms at the beginning of Section 5 — indeed, how many distinctions between base categories should the grammar logic be allowed to make, and what is left to the constraint system? —, one could further simplify the setting so that an object $d \in D^{\mathcal{A}}$ represents a property $\{ \tau \in S^+ \mid \tau \, F \, d \}$ of strings, via a single $F \subseteq S^+ \times D^{\mathcal{A}}$.

Syntactically, basic categories b of Lambek's calculus have to be replaced by formulas $b(t)$ with first-order (resp. feature) terms t. Two modes of combininig Lambek's logic with feature logic are investigated, "simple" and "complementary" fibring.

2 Combination by Simple Fibring

This combines "feature *term* logic" with Lambek's (propositional) logic, and is motivated by a possible first-order version of Lambek's calculus. In the first-order case[2] , the atomic formulas $b(t)$ are interpreted as

$$[\![b(t)]\!]^{\mathcal{M},g} = \{\, \tau \in b^{\mathcal{S}} \mid \tau\, F_b\, t^{\mathcal{A},g} \,\} = b(t^{\mathcal{A},g}), \tag{2}$$

and in the feature term case, where terms denote subsets of $D^{\mathcal{A}}$ —with $x^{\mathcal{A},g} = \{g(x)\}$—, as

$$[\![b(t)]\!]^{\mathcal{M},g} = \{\, \tau \in b^{\mathcal{S}} \mid \exists d\, (\tau\, F_b\, d \wedge d \in t^{\mathcal{A},g} \,\} = \bigcup \{\, b(d) \mid d \in t^{\mathcal{A},g} \,\}. \tag{3}$$

For the feature term case, the formula $b(t)$ denotes "the elements of $b^{\mathcal{S}}$ which also have property t in some sense", i.e. via the fibring relations F_b. Note that this vage intuitition of what $\tau \in b(t)$ could mean cannot completely motivate (3). It allows strings $\tau \in [\![b(t)]\!]^{\mathcal{M},g} \cap [\![b(\neg t)]\!]^{\mathcal{M},g}$, which both have property t and $\neg t$ "in some sense" — a strange notion of 'having a property' for which I see no need in grammatical descriptions, not even to model ambiguity.[3]

Now, using the interpretation (3) of atomic formulas with feature-terms, the analogy to the first-order Lambek calculus runs into both a syntactic and a semantic problem, if one tries to support proof search by unification.

The syntactic problem is this. At first sight, it seems conceivable that feature-graph unification can play the same role in such a system as first-order unification does in resolution-based proof systems for the first-order case. For example, – cf. the discussion following Theorem 3 – a mismatch between an argument and the antecedent of a functor category can be resolved by feature unification. As in first-order logic, this would turn the goal

$$n\left(\; num : sg \;\right),\; n\begin{pmatrix} num : sg \\ pers : 3rd \end{pmatrix} \to s\left(\; true \;\right) \rhd s\left(\; true \;\right)$$

into

$$n\begin{pmatrix} num : sg \\ pers : 3rd \end{pmatrix},\; n\begin{pmatrix} num : sg \\ pers : 3rd \end{pmatrix} \to s\left(\; true \;\right) \rhd s\left(\; true \;\right)$$

and lead to the axiom $s(true) \rhd s(true)$ using $\to$-rules, although we do not use instantiation of variables to refine the feature terms envolved. Thus, feature unification sometimes *does* turn a goal sequent $U \rhd A$ into a refined provable sequent. It then would be a matter of category subsumption and lexicon entries to check whether the lexical assignment U to atomic parts of the input string "lifts" to the refined assignment or not.

But the example is misleading, because only a local modification of the sequent was needed. What really[4] goes wrong with this idea is the following. While in first-order logic, an instantiation (of terms for variables) of a proof gives another proof (of a specialized claim), a similar property with feature-term

[2] We skip the treatment of quantifiers for the first-order version. The precise notion of feature term is fairly irrelvant.

[3] A restriction to functional relations F_b or the modified definition

$$[\![b(t)]\!]^{\mathcal{M},g} = \{\, \tau \in b^{\mathcal{S}} \mid \forall d\, (\tau\, F_b\, d \to d \in t^{\mathcal{A},g}) \,\}$$

would force $[\![b(t)]\!]^{\mathcal{M},g} \cap [\![b(\neg t)]\!]^{\mathcal{M},g} = \emptyset$ and still make the basic Propositions 3 and 4 true. Maybe definition (3) enters the authors' completeness proofs, of which I did not have a full version.

[4] i.e. instead of the lack of ground terms, as the authors claimed in an earlier version of the paper.

refinements (which are not based on variable substitutions) apparently does not hold — due to the locality of the refinement as opposed to the global refinement by substitution.

To overcome this problem of proof syntax, I propose the following quite natural trick: use structure sharing to ensure that the proof is linear in the relevant terms, i.e. abstract from different occurrences. That is, we only use atomic formulas $b(x)$ with *variables* as terms, and keep an envioronment $[x_1 := t_1, \ldots, x_n := t_n]$ naming the feature-terms of the former proof by the variables used to replace them. Now if the Lambek-proof needs an identity $x_i = x_j$, apply feature-unification to (t_i, t_j) and let the refined feature-term be the *shared* entry for x_i and x_j in the environment. Thus, all feature term refinements — not just substitutions — are global to the proof, and certainly Lambek-derivations are preserved under such operations, just as they are under substitution of first-order terms.

The second, semantic problem comes dircetly from the meaning of basic formulas $b(t)$ as given in (3). While for each individual $d \in D^{\mathcal{A}}$, the set $b(d)$ is an *arbitrary* subset of $b^{\mathcal{S}}$, the meaning of $b(t)$ *depends monotonically* on the size of the set $t^{\mathcal{A},g} \subseteq D^{\mathcal{A}}$. Hence, the semantic validity

$$\models \; b(t \wedge s) \; \triangleright b(t) \tag{4}$$

has to be reflected in the proof rules, as is done in the paper. But note that (4) is already in conflict with a unification-based proof system, which turns any goal $b(t_1) \; \triangleright b(t_2)$ into an axiom of the form $b(t) \; \triangleright b(t)$ with *syntactically identical* terms.

Can we fix this problem as well — giving up definition (3), of course, whose motivation wasn't terribly convincing anyway? Let us step back and consider why a first-order version of Lambek's calculus *does* allow unification-based proof methods. Basically, it is because there is no other notion of equality between terms involved but syntactic identity, and the method (of substitution) to refine different terms to the same one can be used to refine proofs as well.

To put it differently, in the first-order case we have Lambek's calculus, except that instead of finitely many basic categories b we have infinitely many $b(t)$. But *the notion of proof does not change* (until we add quantifiers), only that instead of an axiom $b \; \triangleright b$ we have all the $b(t) \; \triangleright b(t)$, etc. — the same b has to be replaced by the same $b(t)$ everywhere in a proof. What we gain are proofs that are schematic in their free individual variables. In particular, free individual variables in a sequent are read universally quantified (with the whole sequent as scope) – an important difference to what the authors do with variables in their second method of 'complementary' fibring.

If we want, *of course* we can have the same with feature terms. But then the semantics must faithfully respect the syntactic notion of identity of terms. For example, we need

$$\models \; b(t) \; \triangleright b(s) \quad \Longleftrightarrow \quad \models \; t = s, \tag{5}$$

which contradicts (4). For the fibred structures $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ this means that we must have $F_b \subseteq 2^{D^{\mathcal{A}}}$ instead of $F_b \subseteq D^{\mathcal{A}}$, so that each subset $C \subseteq D^{\mathcal{A}}$ defines a property $b(C) \subseteq b^{\mathcal{S}}$ in an *arbitrary* way. It then no longer matters whether we use individuals d or sets C of individuals to parameterize the basic categories.

The system sketched above, with an environment of shared feature terms and only variables instead of terms in basic formulas, would fit to such a semantics (if we perform a substitution $[x_i/x_j]$ on the proof when unifying the corresponding terms in the environment).

In this technical sense, I believe it is quite simple to have the perfect analog to a (quantifier-free) first-order version of Lambek's calculus: we only have to change the syntax of proofs a bit to get shared terms and replace first-order by feature-graph unification. However, whether any of these systems is the right one for the intended grammatical descriptions, is another matter.

3 Combination by Complementary Fibring

The "simple fibring" mode of combining feature logic and Lambek's logic — adding feature terms as arguments to basic categories in the way as is done in the paper — does not allow a proof system consisting of Lambek's calculus and unification. The reaction of the authors is to propose another mode of combination, called "complementary fibring", which is a combination of Lambek's logic with "feature *constraint* logic".

Syntactically, atomic formulas b of Lambek's logic now are replaced by formulas $b(x)$, i.e. unary predicates applied to variables (as first-order terms). Feature constraint formulas concerning these variables are imposed as global side conditions. The intention[5] is that feature term refinements are described by the constraints, and the replacement of terms by variables in the formulas is to guarantee that refinements operate globally in a proof.

From the proof theoretic point of view, this combination of Lambek's and feature logic is designed in a way that resembles fairly well current implementations of grammar formalisms that are based on an enrichment of logic programming by appropriate data structures. In fact, the aim of this approach is:

> "We want as solutions for a given goal G those consistent feature constraints φ for which it holds that all fibred models $\mathcal{M}$ of φ are also models of G." (6)

The way to achieve this is to replace atomic formulas b in Lambek's calculus by formulas $b(x)$ with *variables* x, interpreted in fibred structures $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ as

$$[\![b(x)]\!]^{\mathcal{M},g} = \{\, \tau \in b^{\mathcal{S}} \mid \tau\, F_b\, g(x) \,\}. \tag{7}$$

Instead of sequents $A_1, \ldots, A_n \rhd A_0$ one considers annotated sequents G of the form

$$\varphi_1 :: A_1, \ldots, \varphi_n :: A_n \rhd \varphi_0 :: A_0, \tag{8}$$

where the A_i are categories and the φ_i feature constraint formulas. Such an annotated sequent G is satisfied in $\mathcal{M}$ under an assignment g if

$$\mathcal{A}, g \models \varphi_0 \wedge \ldots \wedge \varphi_n \quad \text{and} \quad [\![A_1]\!]^{\mathcal{M},g} \cdots [\![A_n]\!]^{\mathcal{M},g} \subseteq [\![A_0]\!]^{\mathcal{M},g}. \tag{9}$$

The authors then give a complete proof system that determines from a goal sequent G those consistent feature constraint formulas ψ whose fibred models $\mathcal{M}, g$ (i.e. where $\mathcal{A}, g \models \psi$) satisfy G. These ψ constitute the *solutions* of the goal sequent G. In this way, aim (6) is fulfilled, but *why is this the right aim*?

The reader has to speculate in what sense the authors intend to use these notions to define a language. My first guess was the following. Let LA be a lexical assignment over the string structure $\mathcal{S}$, whose entries $v : (\varphi :: A)$ are words $v \in S^+$ paired with categories A and feature constraints φ. Presumably we have $\mathit{free}(\varphi) \subseteq \mathit{free}(A)$ and no two $(\varphi :: A)$ have variables in common, since different lexical entries don't know about each other. Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{F})$ be a fibred structure that satisfies the lexicon in the sense that for all $v : (\varphi :: A) \in LA$, we have $\mathcal{A}, g \models \varphi$ and $v \in [\![A]\!]^{\mathcal{M},g}$. One is tempted to think *the language defined by LA in its model $\mathcal{M}, g$*, with respect to a possibly complex category $S(x_1, \ldots, x_n)$, is

$$L(\mathcal{M}, g) := \left\{ v_1 \cdots v_n \;\middle|\; \begin{array}{l} n \in I\!N,\ v_1 : (\varphi_1 :: A_1), \ldots, v_n : (\varphi_n :: A_n) \in LA, \\ \mathcal{M}, g \text{ satisfies } \varphi_1 :: A_1, \ldots, \varphi_n :: A_n \rhd \varphi_S :: S \end{array} \right\} \tag{10}$$

and then let the language defined by LA be the intersection of the languages of its models, for example. But this was mislead by the paper's emphasis on the fibred structures. In fact, no intuitive underlying

[5] According to personal communication with J.Dörre, the combination I proposed above is equivalent to the "complementary fibring"; but I am not totally convinced that it is.

notion of semantic validity for annotated sequents is used in the authors' construction. (In particular, the feature constraints on the resulting category need not be implied by those on the assumed categories.)

Apparently, it was rather the possible combinations of proof strategies for Lambek's calculus and constraint solving that steered the "complementary fibring" idea. From this proof theoretic perspective, *the language defined by a lexicon LA* ought to be

$$L(LA) := \left\{ v_1 \cdots v_n \;\middle|\; \begin{array}{l} n \in I\!N,\ v_1 : (\varphi_1 :: A_1), \ldots, v_n : (\varphi_n :: A_n) \in LA, \\ v_1 : (\varphi_1 :: A_1), \ldots, v_n : (\varphi_n :: A_n) \ \triangleright \varphi_S :: S \text{ is provable} \end{array} \right\}, \tag{11}$$

in analogy to the notion for Lambek categorial grammars. But how can we rely on this analogy if we do not work with string sets $[\![\varphi :: A]\!]^{\mathcal{M},g}$ here that are combined using the connectives of Lambek's system with their standard semantics?

Inspecting the authors' proof system, we find Lambek's calculus (with atomic categories $b(x)$) and the 'linking' rule

$$\frac{(A_1, \ldots, A_n \ \triangleright A_0)\sigma}{\varphi_1 :: A_1, \ldots, \varphi_n :: A_n \ \triangleright \varphi_0 :: A_0}, \quad \text{if } (\varphi_0 \wedge \ldots \wedge \varphi_n)\sigma \text{ is consistent}, \tag{12}$$

where σ is a substitution of variables by variables. The authors' operational reading of this rule is reasonable: to proof the annotated goal sequent, try to prove the sequent $A_1, \ldots, A_n \triangleright A_0$ with Lambek's calculus; in doing so, certain variables occurring in atomic categories have to be identified, which gives a substitution σ and, possibly, a proof of the specialized sequent $(A_1, \ldots, A_n \ \triangleright A_0)\sigma$. The constraints φ_i on previously distinct variables x now are strengthened constraints $\varphi_i\sigma$ on possibly identical variables $x\sigma$. The consistency check tests whether parameterizing feature 'objects' satisfying the accumulated constraints exist.

Admittedly, this is a combination of feature constraint logic with Lambek's logic, where unification is used to specialize a derivation in the Lambek calculus to a proof. But the combination of the two logics is very loose. Annotated sequents

$$\varphi_1 :: A_1, \ldots, \varphi_n :: A_n \ \triangleright \varphi_0 :: A_0$$

can be seen as pairs $\langle \varphi, G \rangle$ of a sequent G of a calculus for a first-order logic (not necessarily Lambek's) and a constraint formula φ (not necessarily about features). Nothing — if not through the lexicon! — hangs on the fact that the constraint is obtained as a conjunction of constraints in the assumptions and the right hand side formula. The constraint just acts as a filter that refutes some otherwise provable sequents; it is not intimately connected with the logic. Hence, this "complementary fibring" is possible for any subsystem of first-order logic and an arbitrary constraint logic about terms that has a decidable satisfyability check.

What I find irritating about rule (12) is that it can only occur as the final rule in a proof, because the annotated formulas $\varphi_i :: A_i$ are not allowed in Lambek's rules. Also, I miss an intuitive semantics according to which the rule is sound. (Do you see this from the completeness proof?). One could imagine a variation

$$\frac{(A_1, \ldots, A_n \ \triangleright A_0)\sigma}{(\varphi_1 :: A_1, \ldots, \varphi_n :: A_n \ \triangleright \varphi_0 :: A_0)\sigma}, \quad \text{if } (\varphi_0 \wedge \ldots \wedge \varphi_n)\sigma \text{ is consistent}, \tag{13}$$

– and here the σ could as well be omitted. This would be sound and express a 'specialization', under

$$(\varphi :: A)^{\mathcal{M}} := \bigcap \{ \tau \in A^{\mathcal{M},g} \mid g : \mathit{Var} \to D^{\mathcal{A}},\ \mathcal{A}, g \models \varphi \}. \tag{14}$$

Note also that using (14), lexical entries $\tau : (\varphi :: A)$ get a *polymorphic* interpretation: τ has *all the* categories $A(d_1, \ldots, d_k)$ where the parameters $d_1, \ldots, d_k$ satisfy the condition $\varphi(x_1, \ldots, x_k)$. If we reinterprete the connectives according to

$$(\varphi :: A) \cdot (\psi :: B) := (\varphi \wedge \psi) :: (A \cdot B), \qquad (\varphi :: A) \to (\psi :: B) := (\varphi \wedge \psi) :: (A \to B),$$

we would obtain $(\varphi :: A)^{\mathcal{M}} \cdot (\varphi :: A \to \psi :: B)^{\mathcal{M}} \subseteq \psi^{\mathcal{M}}$ for example, and so it might be that one could obtain a system like Lambek's, but with a modular treatment of the constraints spread over the rules for the various connectives. How this relates to the computationally more efficient system alluded to in the paper, I cannot say. But note that because of

$$\models \varphi(x) :: b(x) \triangleright \psi(x) :: b(x) \quad \Longleftrightarrow \quad \varphi \models \psi, \tag{15}$$

we again would have a monotonicity built into the logic, and could not simply use Lambek's rules and unification.[6] Since this is speculation that needs substantiation by further research, better let us return to the paper.

Accepting the authors' aim (6), the completeness theorem they present gives a semantics to their proof system, and hence to some current grammar formalisms that combine (i) a proof system for a consequence relation in the area of linguistic reasoning with (ii) methods of constraint solving to check additional constraints on proofs. In that sence, in spite of all critical remarks so far, the paper constitutes a real contribution to open problems. It may also well be that the complexity results, which I did not check carefully, give a better understanding of the computational aspects of some current systems.

However, I must confess that I did not fully understand the intuitive motivation behind the "complementary fibring", nor did I find the model theoretic semantics for this approach illuminating. This is probably because the completeness result is not about validity in the expected sense. Maybe the mixture of consistency (of relations between category names) and consequence (as inclusions between named string sets) in the definition is strange only in the general case, but more natural when implicit assumptions were added. These would have become apparent via a presentation of linguistic examples!

4 Comments

Let me make two more fundamental remarks that I find are to be considered in building a "logic" of reasoning with features.

4.1 Approximation

Somehow I did expect that the notion of fibred structure comes with an approximation relation $d \sqsubseteq d'$ between elements of its feature structure $\mathcal{A}$. This relation ought to be reflected by the fibring of basic syntactic categories, in the sense that

$$d \sqsubseteq d' \in D^{\mathcal{A}} \;\Rightarrow\; [\![b(d)]\!]^{\mathcal{M}} \subseteq [\![b(d')]\!]^{\mathcal{M}}. \tag{16}$$

For example, what happens with $[\![b(x)]\!]^{\mathcal{M},g} = b(g(x))$ if the constraints on x approach inconsistency – does $g(x)$ *approach* nonexistence, and $b(g(x))$ the empty set? Since we don't habe (16), both in the simple and the complementary fibring combination there is *no systematic connection from relations between the feature objects $d \in D^{\mathcal{A}}$ to relations between the sets $b(d)$ of strings they are correlated with.* The feature structure just provides some *names* to denote sets of strings. I am unable to see why a theory of grammar, as inclusion statements between sets of strings, should care about inconsistency of relations between the names of these sets — when the naming is arbitrary.

4.2 Parametrization of basic categories

In a sense, a similar question affects the first-order version too: what is the parametrization of $b^{\mathcal{S}}$ into subsets *by means of terms* good for, compared with, say, using additional predicates? Somehow a

[6] Bounded quantification $\forall d \in \varphi.\, \tau \in A(d)$ simply is different from unbounded quantification $\forall d.\, \tau \in A(d)$.

linguistic motivation is needed to justify the *specific kind of terms used*, because the terms now say what a 'legal' category name is, and this can hardly be totally independent of the denotations of the categories.

The authors' hints that (feature) terms encode "partial information" and "provide a general means for structuring data" as in logic programming could be more specific. In my view, the main point in having the $b(t)$'s is that the set of basic categories now becomes infinite — which is hardly needed —, and partially ordered by the subsumption relation of their term parts.

The puzzle is what it *logically* means to use category names $b(t)$ structured according to a particular kind of terms. (Note that the structure of category names in categorial grammar is *directly* reflected in the string semantics.) The only answer I see is that the term structure gives the possibility of grammar rule *schemata*, if rules may have free individual variables (which implicitly are universally quantified) appearing in the terms of the complex category names.

The structure of terms allowed determines *what* can be expressed schematically, and hence needs some motivation. For example, abstraction over linguistically reasonable classes of predicates ought to be expressible as abstraction from part of the term structure in grammar rules. If there is a logical advantage in using feature terms rather than first-order terms, it ought to be related to the possibility of abstraction over refinements, a kind of subtype quantification.

But does all this make sense unless we assume that the hierarchical organization of basic category *names* $b(t)$ corresponds to a similar organization of the string sets $b(t)^{\mathcal{M},g}$ denoted? Otherwise, wouldn't it amount to an endless reorganization of category names, if linguistic abstractions just had to be squeezed into a particular discipline of stating generalizations as schematic in the names of string sets?

In contrast, the semantics of the combined fibring, based on (7) alone, does not share this view. I do not believe that the hierarchical organization of category names is reasonable without something like (16), i.e. a corresponding hierarchy in the boolean structure of string sets.

5 Summary

The paper uses the idea of fibring to understand possible combinations of Lambek's logic with two versions of feature logics. The authors' preferred combination has a simple-looking proof system that seems to describe some current grammar formalisms fairly well, but its semantics is presented as a mixture of satisfaction and validity that is fairly unfamiliar. I am not convinced that this complication is necessary, and hope that further efforts result in simpler solutions.

The paper[7] suffers from a missing analysis of what the *logical* status of features in grammatical descriptions really is, and how partial information is used in language specification (cf. section 4.2) and parsing. Without such an analysis, I think, there is no way to answer convincingly what kind of combination between feature "logic" and Lambek's logic is to be aimed at. A presentation of clear intuitive aims to be established by such combinations is needed, and could also give criteria for success or failure of the various combinations.

My personal problem with this paper is not about technical soundness and accuracy, but about lack of motivation, examples and preciseness of goals. There were several design decisions I found incomprehensible — for example, why are only *basic* categories parameterized? —, and some discussions of alternatives that I found incomplete. The guiding ideas ought to be brought out clearer in a final version, and hopefully the proposals made in the comment are judged as a support in this direction, and not as mere criticism.

Acknowledgement: I would like to thank Martin Emms for several discussions about these matters.

[7]In the versions I have seen. Some of the critical remarks here may not apply to the final version; to escape from a loop of comments and improvements, I did not cross-check with the latest version.